

# Tema 1

**Aplicaciones nativas:** Se desarrollan usando su lenguaje nativo específico. No se pueden usar en otro Sistema Operativo distinto a aquel para el que fueron programadas.

**Aplicaciones nativas multiplataforma:** Se desarrollan usando un lenguaje intermedio, mezclan conceptos nativos de cada plataforma y comunes, y generan una aplicación para cada plataforma (Ej: Maui o Flutter).

**Aplicaciones híbridas con html5:** Son multiplataforma, se genera una aplicación distinta para cada plataforma, pero es en realidad un navegador web funcionando como si fuese una aplicación, por lo que no usa conceptos nativos, sobre todo de interfaz (Ej: Angular con Cordova y ReactJS).

## Ventajas del desarrollo nativo:

- Rendimiento más alto.
- Acceso al Hardware dado directamente por el sistema.
- Posibilidad de sacar más partido a las novedades de las nuevas versiones del sistema.

## Es recomendable para situaciones en que:

- El rendimiento alto es esencial.
- Se requieren procesar grandes cantidades de datos in-app.
- Se quieren gestionar recursos del sistema o sistemas operativos.
- Se quieren programar apps con muchos vídeos/videojuegos.

## Desventajas del desarrollo nativo:

- Si se soportan múltiples SO móviles, se tiene que desarrollar una app por plataforma.
- El tiempo de testeo y mantenimiento se ven afectados, pues se necesitan probar varias apps.

## Ventajas del desarrollo nativo multiplataforma:

- Gran parte del código se puede compartir para todas las plataformas.
- La interfaz de usuario se genera utilizando controles nativos.
- Tiempo de desarrollo menor que el de todas las apps nativas.

## Se recomienda su uso cuando:

- Se puede renunciar a un alto rendimiento de la app y se necesita un coste de desarrollo relativamente bajo. Compromiso entre el desarrollo nativo y el desarrollo híbrido con HTML5.

### **Desventajas del desarrollo nativo multiplataforma:**

- No se comparte todo el código, se tiene que escribir algo de código nativo.
- La velocidad de ejecución se puede ver mermada.
- El acceso a las características del dispositivo y el SO dependen del framework y sus plugin, adaptación más lenta a nuevas características.
- La personalización de la interfaz de usuario depende del soporte del framework.

### **Ventajas del desarrollo híbrido HTML5:**

- La principal: Se desarrolla muy rápido con este tipo de plataformas, permitiendo hacer las cosas más comunes en multiplataforma sin mucho esfuerzo. Es la que saca partido al sistema de forma menos eficiente, y la de rendimiento más bajo.
- El coste es mucho más bajo.

### **Se recomienda usarlo cuando:**

- La app no usa capacidades demasiado específicas del sistema.
- El desarrollo tiene que ser rápido y con un coste bajo.

### **Desventajas del desarrollo híbrido HTML5:**

- Al ser basadas en web y usar por debajo un navegador, hay más posibilidades de que haya fallos de seguridad.
- La velocidad se puede ver mermada, por la interacción con el controlador web.
- El acceso a las características del dispositivo y el SO dependen del soporte del framework y sus plugin.
- No se pueden utilizar controladores específicos de la interfaz de la plataforma.

## Tema 2

### Conceptos iniciales (Los que tienen la letra más grande son los que van a caer)

- **Frame** es cada uno de los “dibujos” estáticos que se hacen en pantalla y corresponden con la representación gráfica de la parte visible del mundo.
- **FPS** (Frames per Second) es la cantidad de Frames que un motor es capaz de dibujar por segundo en un dispositivo. La rápida sucesión de Frames da la sensación de movimiento en el juego.
- **Delta Time** es el tiempo que transcurre entre el dibujado de dos Frame. Normalmente expresado en segundos.
- **Render** (Renderizar) es la acción de dibujar un Frame en pantalla.
- **Textura** se refiere a cada una de las imágenes 2D que el ordenador tiene que cargar en memoria para renderizar un frame que contenga textura. Aunque las texturas sean en 2D pueden renderizar en objetos 2D o renderizar en objetos 3D.
- **Sprite** es la representación 2D de una textura en una posición y con una capacidad de colisión determinada por una Hitbox.
- **Hitbox** (Caja de colisión) es una forma invisible (podemos hacerla visible para debuggear) que determina los límites de colisión de un objeto. La más básica es rectangular, aunque puede tener cualquier forma.
- **Animación** es el conjunto de texturas 2D, normalmente almacenadas en un Atlas. Cada textura representa la visualización en un Frame de una textura en movimiento.
- **Atlas** es un conjunto de texturas que se usan juntas o muy frecuentemente, empaquetadas en un mismo archivo, para evitar muchas cargas desde disco a memoria. Pueden ser animaciones o no serlo. Hay que escoger un equilibrio entre no tener atlas y cargarlo todo desde disco (cargas individuales muy rápidas, funcionamiento global ralentizado), o tener solo atlas con todas las texturas del juego (carga inicial muy lenta, funcionamiento global muy rápido).
- **Asset** es un elemento individual del juego. Puede referirse a una textura, un sonido, un modelo 3D...
- **HUD** (Head-up Display) Información del juego que se representa por encima del dibujo del mundo. Cosas como puntuación o número de anillos en un juego de Sonic, el inventario rápido en Minecraft, o la lista de armas disponibles en un Shooter como Unreal Tournament.

## Tipos de juegos

- **JFPG:** Juego de Rol Japonés (Japanese Role-Playing Game), son un subgénero de los juegos de rol (RPG) que se originaron en Japón. Estos juegos suelen tener elementos de narrativa rica, personajes memorables y sistemas de combate por turnos o en tiempo real. Ejemplos populares incluyen la serie “Final Fantasy” y “Dragon Quest”.
- **Juegos de lucha:** Esta categoría se centra en enfrentamientos uno contra uno o en equipos donde los jugadores controlan personajes que se enfrentan en combates cuerpo a cuerpo. Ejemplos famosos son “Street Fighter”, “Mortal Kombat” y “Super Smash Bros”.
- **Juegos de acción/aventura:** Este género combina elementos de acción y aventura en un solo juego. Los jugadores suelen explorar un mundo abierto. Resolver acertijos y participar en combates en tiempo real. Ejemplos incluyen la serie “The Legend of Zelda” y “Assassin’s Creed”.
- **RPG occidentales:** Son juegos de rol desarrollados en países occidentales, como EEUU y Europa. Suelen enfocarse en la personalización de personajes, elecciones morales y mundos abiertos. Ejemplos populares son “The Elder Scrolls” y “The Witcher”.
- **RTS:** Estrategia en tiempo real (Real-Time Strategy), son juegos de estrategia en tiempo real y son aquellos en los que los jugadores toman decisiones estratégicas en tiempo real y controlan ejércitos o recursos para lograr objetivos. Ejemplos incluyen “Age of Empires” y “StarCraft”.

## Tema 3

### Términos Unity

- **GameObject:** Cualquier elemento que aparece en una escena: Cámara, luces, textos, objetos... Es la entidad básica de Unity. Los GameObject contienen Components.
- **Components:** Elementos que determinan el comportamiento del GameObject, configuraciones como el tamaño, posición, colisionadores...
- **Prefab:** GameObject especial que hace “clones” de objetos que tengamos definidos. Estos objetos Prefab, comparten muchas de las propiedades entre si, salvo la posición. De esta forma, si cambiamos una propiedad en uno, se cambia en todos. Sirven para cuando queremos hacer muchos objetos, por ejemplo árboles de un bosque.
- **BoxCollider:** Colisionador de caja que define un área con forma de cubo alrededor del objeto.
- **Rigidbody:** Permite el comportamiento físico para un objeto, inmediatamente responderá a la gravedad.

Va a preguntar para qué sirve cada código que nos va a poner (van a ser códigos que ya hemos hecho y cosas sencillas y fáciles de distinguir)

## Qué es el inspector

Es una herramienta que nos ayuda a modificar objetos. Sus opciones dependen del objeto seleccionado.

## ¿Cómo se importa y exporta un paquete en Unity? ¿Y un ejecutable?

[19:41, 11/16/2023] Javi: Le das a assets arriba y para importar le das a importar assets, custom package y si quieres exportar pues a exportar assets

[19:42, 11/16/2023] Javi: Y el ejecutable en File le das a build

[19:42, 11/16/2023] Javi: Y ya pos le das a windows para los pobres y mac para privilegiados y te hace una aplicación

Gracias Javi por tu aporte.

## ¿Qué es un vector 3, qué hace el transform position?

Transform, es una propiedad de GameObject que permite hacer movimientos directos sin cambios de orientación, a partir de un Vector3, que nos indica una dirección tridimensional.